BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－161933

(43)公開日　平成10年(1998)６月19日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　12/14	３１０	Ｇ０６Ｆ　12/14	３１０Ａ
9/46	３４０	9/46	３４０Ｆ

審査請求　未請求　請求項の数24　ＯＬ　（全 10 頁）

(21)出願番号　特願平9－254505

(22)出願日　平成９年(1997)９月19日

(31)優先権主張番号　０８／７２１１４５

(32)優先日　1996年９月26日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
INTERNATIONAL BUSINESS MASCHINES CORPORATION
アメリカ合衆国10504、ニューヨーク州アーモンク　（番地なし）

(72)発明者　ドナルド・フレッド・オールト
アメリカ合衆国12538　ニューヨーク州ハイド・パーク　ルーズベルト・ロード　115

(74)代理人　弁理士　坂口　博　（外１名）

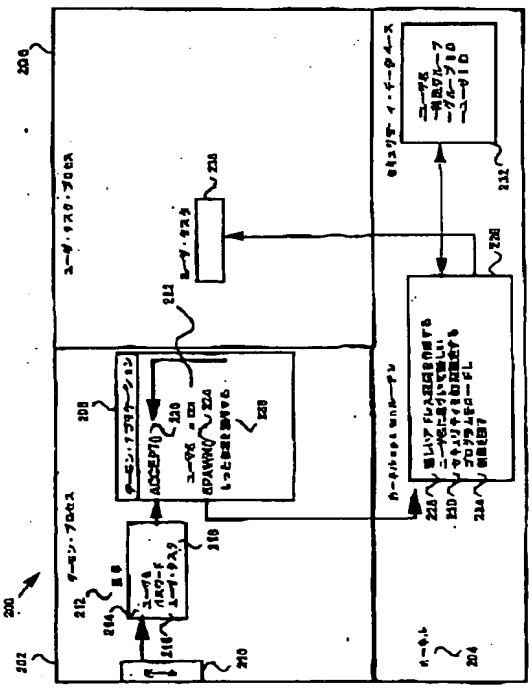
最終頁に続く

(54)【発明の名称】　クライアント／サーバ・システムにおける適切なセキュリティ環境でのユーザ・タスク実行方法及び装置

(57)【要約】

【課題】　クライアント／サーバシステムにおいて、監視デーモンが指定されたタスクをユーザに代わって実行することができるようにする方法及び装置を提供する。

【解決手段】　監視デーモンは、ユーザ要求を受け取ると、要求で指定されているユーザ・アイデンティティに従って環境変数を設定し、オペレーティング・システム・カーネルに対してシステム・コールを出して要求で指定されているユーザ・タスクを作成する。オペレーティング・システム・カーネルは、このシステム・コールに応答して指定されたユーザ・タスクのための新しいアドレス空間を作成し、環境変数に従ってユーザ・タスクのためのセキュリティ環境を作成してからユーザ・タスクを新しいアドレス空間で開始する。



1

【特許請求の範囲】

【請求項１】要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視するサーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わって前記タスクを実行する方法であって、

（ａ）デーモン・プロセスが、前記ユーザからの前記要求を受け取ると、［１］前記要求で指定された前記アイデンティティに従って環境変数を設定するステップと、［２］前記オペレーティング・システム・カーネルに対してシステム・コールを出して前記指定されたユーザ・タスクを新しいアドレス空間内で実行するステップと、

（ｂ）前記オペレーティング・システム・カーネルが前記デーモン・プロセスから前記システム・コールを受け取ると、［１］前記指定されたユーザ・タスクのために新しいアドレス空間を作成するステップと、［２］前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成するステップと、［３］前記新しいアドレス空間内で前記指定されたユーザ・タスクを開始するステップとを含む方法。

【請求項２】前記オペレーティング・システム・カーネルがＰＯＳＩＸ準拠オペレーティング・システム・カーネルであることを特徴とする、請求項１に記載の方法。

【請求項３】前記システム・コールがｓｐａｗｎ（）システム・コールであることを特徴とする、請求項１に記載の方法。

【請求項４】前記ユーザがユーザ名を有し、前記アイデンティティが前記ユーザ名を含むことを特徴とする、請求項１に記載の方法。

【請求項５】前記環境変数が前記ユーザ名と等しく設定されることを特徴とする、請求項４に記載の方法。

【請求項６】前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザＩＤを有し、前記セキュリティ環境を作成する前記ステップ（ｂ）［２］が、前記新しいユーザ・アドレス空間のユーザＩＤを、前記環境変数によって指定されたユーザ名のユーザＩＤと等しく設定することを特徴とする、請求項５に記載の方法。

【請求項７】前記ユーザ名のユーザＩＤを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、請求項６に記載の方法。

【請求項８】前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループＩＤを有し、前記セキュリティ環境を作成する前記ステップ（ｂ）［２］が、前記新しいアドレス空間のグループＩＤを、前記環境変数によって指定されたユーザ名のグループＩＤと等しく設定することを特徴とする、請求項５に記載の方法。

2

【請求項９】要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視する前記サーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わって前記タスクを実行する装置であって、

（ａ）前記デーモン・プロセスに関連づけられ、前記ユーザからの前記要求の受取りに応答して、［１］前記要求に指定された前記アイデンティティに従って環境変数を設定し、［２］前記オペレーティング・システム・カーネルに対してシステム・コールを出して新しいアドレス空間内で前記指定されたユーザ・タスクを実行する手段と、

（ｂ）前記オペレーティング・システム・カーネルに関連づけられ、前記デーモン・プロセスからの前記システム・コールの受取りに応答して、［１］前記指定されたユーザ・タスクのために新しいアドレス空間を作成し、［２］前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成し、［３］前記新しいアドレス空間内で前記指定されたユーザ・タスクを開始する手段とを含む装置。

【請求項１０】前記オペレーティング・システム・カーネルがＰＯＳＩＸ準拠オペレーティング・システム・カーネルであることを特徴とする、請求項９に記載の装置。

【請求項１１】前記システム・コールがｓｐａｗｎ（）システム・コールであることを特徴とする、請求項９に記載の装置。

【請求項１２】前記ユーザがユーザ名を有し、前記アイデンティティが前記ユーザ名を含むことを特徴とする、請求項９に記載の装置。

【請求項１３】前記環境変数が前記ユーザ名と等しく設定されることを特徴とする、請求項１２に記載の装置。

【請求項１４】前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザＩＤを有し、前記セキュリティ環境を作成する前記手段（ｂ）［２］が、前記新しいアドレス空間のユーザＩＤを、前記環境変数によって指定されたユーザ名のユーザＩＤに等しく設定する手段を含むことを特徴とする、請求項１３に記載の装置。

【請求項１５】前記ユーザ名のユーザＩＤを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、請求項１４に記載の装置。

【請求項１６】前記アドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループＩＤを有し、前記セキュリティ環境を作成する前記ステップ（ｂ）［２］が、前記新しいアドレス空間のグループＩＤを、前記環境変数によって指定されたユーザ名のグループＩＤに等しく設定することを特徴とする、請求項１３に記

載の装置。

【請求項17】要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・システム・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視する前記サーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わってタスクを実行する方法ステップを行う機械によって実行可能なプログラムを有形に実施する機械可読プログラム記憶装置であって、前記方法ステップが、

（a）前記デーモン・プロセスが、前記ユーザからの前記要求を受け取ると、［1］前記要求に指定された前記アイデンティティに従って環境変数を設定するステップと、［2］前記オペレーティング・システムに対してシステム・コールを発行して前記指定されたユーザ・タスクを新しいアドレス空間内で実行するステップと、

（b）前記オペレーティング・システム・カーネルが、前記デーモン・プロセスから前記システム・コールを受け取ると、［1］前記指定されたユーザ・タスクのために新しいアドレス空間を作成するステップと、［2］前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成するステップと、［3］前記指定されたユーザ・タスクを前記新しいアドレス空間内で開始するステップとを含む、プログラム記憶装置。

【請求項18】前記オペレーティング・システム・カーネルがＰＯＳＩＸ準拠オペレーティング・システム・カーネルであることを特徴とする、請求項17に記載のプログラム記憶装置。

【請求項19】前記システム・コールがｓｐａｗｎ（）システム・コールであることを特徴とする、請求項17に記載のプログラム記憶装置。

【請求項20】前記ユーザがユーザＩＤを有し、前記アイデンティティが前記ユーザＩＤを含むことを特徴とする、請求項17に記載のプログラム記憶装置。

【請求項21】前記環境変数が前記ユーザＩＤと等しく設定されることを特徴とする、請求項20に記載のプログラム記憶装置。

【請求項22】前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザＩＤを有し、前記セキュリティ環境を作成する前記ステップ（b）［2］が、前記新しいアドレス空間のユーザＩＤを、前記環境変数によって指定されたユーザ名のユーザＩＤと等しく設定するステップを含むことを特徴とする、請求項21に記載のプログラム記憶装置。

【請求項23】前記ユーザ名の前記ユーザＩＤを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、請求項22に記載のプログラム記憶装置。

【請求項24】前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループＩＤを有し、

前記セキュリティ環境を作成する前記ステップ（b）［2］が、前記新しいアドレス空間のグループＩＤを、前記環境変数によって指定されたユーザ名のグループＩＤと等しく設定するステップを含むことを特徴とする、請求項21に記載のプログラム記憶装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、新しいユーザ・アイデンティティと適切な特権を使用してＰＯＳＩＸ環境に新しい作業単位を作成する方法に関する。

【0002】

【従来の技術】（複数の相互接続された機械に作業が分散される）分散コンピューティング・システムは、クライアント／サーバ・モデルに基づいて構築されることが多い。このモデルでは、クライアント・プロセス（または単に「クライアント」）がサーバ・プロセス（または単に「サーバ」）に対して、プリント・サーバの場合はファイルの印刷、ファイル・サーバの場合はファイルの取り出しまたは記憶、あるいはアプリケーション・サーバの場合はアプリケーションの実行など、指定したサービスを実行するように求める要求を出す。クライアント・プロセスとサーバ・プロセスとは同じ物理機械上に存在することができるが、両者は異なる機械上に存在するのが一般的であり、以下の説明においてもそうである。サーバはローカル・エリア・ネットワーク（ＬＡＮ）などのほか、インターネットなどのより広域のネットワークでも使用される。本明細書で特に扱うインターネット・サーバの１つのクラスは、ワールド・ワイド・ウェブにサービスを提供するサーバである。ワールド・ワイド・ウェブは、グラフィカル・コンテンツ（「ウェブ・ページ」）を提供し、ハイパーテキスト転送プロトコル（ＨＴＴＰ）に従ってその他のサービスを行うインターネット・サイトの集まりである。

【0003】ワールド・ワイド・ウェブ（または単に「ウェブ」）上のサーバなどのサーバは、オペレーティング・システムと共に機能し、オペレーティング・システムはサーバが存在する機械上でシステム資源を管理し、基本システム・サービスを行う。これらのオペレーティング・システムは、ＵＮＩＸオペレーティング・システムまたはＵＮＩＸベースのオペレーティング・システムであることが多い。様々なＵＮＩＸオペレーティング・システムの急増により、そのようなシステムによって提供される共通のサービスのセットを定義する様々な努力がなされてきた。このような１つの努力は、１９８８年に初めて発表されたＩＥＥＥ　ＰＯＳＩＸ　１００３．１仕様と、ＩＥＥＥ　ＰＯＳＩＸ　１００３．１ｄなどの補遺である（以下まとめて「ＰＯＳＩＸ」と呼ぶ）。

【0004】ＰＯＳＩＸ準拠システムでは、ユーザによ

るファイルなどへのアクセスは、各プロセス及び各ファイルに、ユーザのアイデンティティを指定するユーザＩＤとグループＩＤを関連づけることによって制御する。各ファイルには、３つの３ビット・フィールドも関連づけられ、各フィールドはｒｗｘビットと呼ばれる３ビットの許可ビットから成る。この３つのフィールドはそれぞれ、ファイルの所有者と、所有者のグループ内の他のユーザと、所有者のグループに入っていないその他のユーザとに対して許可されたアクセスを定義する。各３ビット・フィールド内で、ｒビットはファイルの読み取りができるかどうかを指定し、ｗビットはファイルの書込みができるかどうかを指定し、ｘビットはファイルの実行ができるかどうかを指定する。プロセスがファイルへのアクセスを要求するときは必ず、プロセスとファイルのユーザＩＤとグループＩＤと、そのファイルのために指定された許可ビットを検査して、そのプロセスを実行しているユーザが要求するアクセスを行うことができるかどうかが判断される。このアクセス制御手続きは、当技術分野で周知であり、Ｗ．Ｒ．スティーブンズ（Ｓｔｅｖｅｎｓ）の「ＵＮＩＸ　Ｎｅｔｗｏｒｋ　Ｐｒｏｇｒａｍｍｉｎｇ」（１９９０年）及びＡ．Ｓ．タネンバウム（Ｔａｎｅｎｂａｕｍ）の「Ｍｏｄｅｒｎ　ＯｐｅｒａｔｉｎｇＳｙｓｔｅｍ」（１９９２年）などの参考資料に記載されている。

【０００５】サーバ環境でこのようなアクセス制御を実現するために、ＰＯＳＩＸ準拠オペレーティング・システムは、新しいユーザ・アイデンティティを使用して新しい作業単位と、その新しい作業単位が異なる１組の資源にアクセスすることができる新しいセキュリティ環境とを作成することができるサーバの能力をサポートしなければならない。このタイプの作業は、従来、サーバ・システム内で「デーモン」と呼ばれる背景プログラムによって行われてきた。一般に、デーモンは、デーモンの権限のサブセットを有するユーザに代わって要求を処理し、タスクを実行する。システムのセキュリティを維持するために、タスクはそのユーザの権限を使用して新しい作業単位の下で実行されなければならない。これらの新しい作業単位の作成は、ｆｏｒｋ（）、ｓｅｔｕｉｄ（）、ｓｅｔｇｉｄ（）、ｅｘｅｃ（）などの関数及びその他の関連するＰＯＳＩＸサービスを使用して扱われる。

【０００６】デーモンの例は、ＰＯＳＩＸシステム上のウェブ・サーバである。このようなウェブ・サーバは、ポートでユーザがＰＯＳＩＸファイル・システムから文書を取り出す要求を渡すのを監視するプログラムである。これを行うために、ウェブ・サーバはまずユーザ名とパスワードを検証しなければならない。次に、ウェブ・サーバはｆｏｒｋ（）関数を使用して新しいアドレス空間を作成し、要求されたユーザ・タスクのために別個の環境を設ける。新しいアドレス空間に入ると、ウェブ・サーバはｇｅｔｇｒｏｕｐｓ（）やｓｅｔｇｒｏｕｐｓ（）などの様々なＰＯＳＩＸ関数を使用して正しいセキュリティ環境を作成し、正しい補足グループＩＤを設定した後、ｓｅｔｇｉｄ（）及びｓｅｔｕｉｄ（）を使用して正しいグループＩＤとユーザＩＤを設定しなければならない。補足グループとグループＩＤとユーザＩＤとは、ＰＯＳＩＸ許可の基礎を形成する。最後にウェブ・サーバはシェルのｅｘｅｃ（）を行い、要求された文書を獲得するシェル・スクリプトを実行する。全体的に見て、これは大量の処理オーバーヘッドを要する長くて複雑なプロセスである。

【０００７】

【発明が解決しようとする課題】ｓｐａｗｎ（）サービスは、ｆｏｒｋ（）サービスとｅｘｅｃ（）サービスを結合して１つのコールにするＰＯＳＩＸ関数である（ＩＥＥＥ　ＰＯＳＩＸ　１００３．１ｄに記載されている）。したがって、ＰＯＳＩＸアプリケーションは、新しいアドレス空間をフォークしてから新しいプログラム・イメージを実行する代わりに、単にその新しいプログラムを異なるアドレス空間に作成して、要求側のアドレス空間をコピーするオーバーヘッドを節約することができる。しかし、この場合の問題は、現在のところ、デーモン自体のセキュリティ環境を変更せずに、新しいプログラム・イメージが制御を獲得する前にユーザ・アイデンティティを変更する方法がないことである。したがって、デーモンが要求された文書を獲得するためにシェル・スクリプトを作成しようとすれば、デーモンのユーザ・アイデンティティの下でＰＯＳＩＸファイル・システムへのアクセスが行われることになり、それによって文書を要求しているユーザのＰＯＳＩＸ許可が置き換えられる可能性がある。ｓｐａｗｎ（）を使用することができないため、デーモン・アプリケーションは前述のように、ユーザ・タスク・プロセス層で実行する余分の面倒なソフトウェアを備えなければならない。

【０００８】

【課題を解決するための手段】一般には、本発明は、指定したユーザ・タスクを実行するように求めるユーザからの要求をデーモン・プロセスが監視するサーバ・システムにおいて、ユーザのための適切なセキュリティ環境を使用してユーザに代わってタスクを実行する方法及び装置に関する。デーモン・プロセスは、ユーザから要求を受け取ると、その要求で指定されているユーザ・アイデンティティ（たとえばユーザ名）に従って環境変数を設定し、オペレーティング・システム・カーネルに対して、指定されたユーザ・タスクを新しいアドレス空間で実行するシステム・コールを出す。オペレーティング・システム・カーネルは、ＰＯＳＩＸ準拠カーネルとすることができ、その場合、システム・コールはｓｐａｗｎ（）システム・コールとすることができる。オペレーティング・システム・カーネルはデーモン・プロセスから

システム・コールを受け取ると、指定されたユーザ・タスクのために新しいアドレス空間を作成し、指定されたユーザ・タスクのために環境変数に従って新しいセキュリティ環境を作成し、新しいアドレス空間内で指定されたユーザ・タスクを実行する。

【０００９】具体的には、本発明によると、新しい環境変数（ＵＳＥＲＮＡＭＥ）を作成することによって従来の技術の前述の欠点が解消される。この環境変数が指定されると、ｓｐａｗｎ（）関数によって呼び出されるプログラムが正しいＰＯＳＩＸ許可を使用して制御を獲得する。これを実現するために、ｓｐａｗｎ（）関数を変更して、ｓｐａｗｎ（）関数がこの新しい環境変数を認識し、有効なユーザ名であることを検証した後、指定されたユーザ名のために新しいアドレス空間の初期作成が行われるようにする。新しいユーザ・アイデンティティを使用して実行されると、セキュリティ・データ・ベースに入っているそのユーザのエンティティから残りのＰＯＳＩＸ許可を入手する。新しいアドレス空間とタスクの初期設定が完了すると、ｓｐａｗｎルーチンはその新しいプログラム・イメージを現行セキュリティ環境で開始する。

【００１０】本発明は、デーモン・プロセス環境に変更を加える必要なしに、ユーザ・タスク・プロセスのためのセキュリティ環境を作成する方法を提供する。さらに、デーモン・アプリケーションのどの部分もユーザ・タスク・プロセスで実行する必要がない（もしそうであればユーザ・タスク・プログラム・イメージに制御を渡す前にユーザ・タスク・プロセス許可の変更を必要とすることになる）。また、本発明は、従来のｆｏｒｋ（）とｅｘｅｃ（）の使用をｓｐａｗｎ（）関数の使用で置き換えることができ、それによってｓｐａｗｎ（）関数が本質的にもたらす簡略化とパフォーマンス向上を実現する。

【００１１】

【発明の実施の形態】図１は、従来の実施態様のデーモンを組み込んだコンピュータ・システム１００（物理機械を含む）の例であり、特定のユーザのために要求されたタスクを実行する様々なソフトウェア層間の関係が示されている。これらの層には、デーモン・プロセス層またはアドレス空間１０２と、オペレーティング・システム（ＯＳ）またはカーネル層またはアドレス空間１０４と、後述のようにして作成されるユーザ・タスク・プロセス層またはアドレス空間１０６とが含まれる。物理機械は、たとえばＳ／３９０並列エンタープライズ・サーバなどのＩＢＭ　Ｓ／３９０プロセッサとすることができ、カーネル層１０４はＰＯＳＩＸ準拠ＯｐｅｎＥｄｉｔｉｏｎ構成要素を有するＩＢＭ　ＯＳ／３９０オペレーティング・システムとすることができる。

【００１２】デーモン・プロセス層１０２には、ユーザに代わって動作して遠隔クライアント（図示せず）に結合されたポートまたは通信回線１１０を監視するソフトウェアを含むデーモン・アプリケーション１０８がある。デーモン・アプリケーション１０８は、入力としてユーザ名１１４とパスワード１１６とタスクを指定する識別子１１８とを含む要求１１２をポート１１０を介して受け入れる（ステップ１２０）。

【００１３】ユーザ名（またはログイン名）１１４は、ユーザに固有に関連づけられ、ユーザが自分自身をシステム１００に識別させるために使用する英数字文字列である。システム１００に付随するセキュリティ・データベース１３４内の各ユーザ名１１４のレコード１３６に、ユーザ名に固有に関連づけられた整数ユーザＩＤと、ユーザ名に固有に関連づけられた整数グループＩＤと、ユーザ名に関連づけられた補足グループのリストのほか、ユーザのパスワード及びその他の関係情報が記憶される。ユーザを認証するために、デーモン・アプリケーション１０８はセキュリティ・データベース１３４内の指定されたユーザ名１１４に対応するレコード１３６にアクセスし、そのレコードに入っているパスワードを調べる。指定されたユーザ名１１４のレコード１３６があり、レコード内のパスワードが要求に添付されたパスワード１１８と一致する場合、要求者はその要求者が自称しているユーザであると認証される。

【００１４】デーモン・アプリケーション１０８はユーザ名１１４を検証し、パスワード１１６を認証した後、要求されたタスクを実行する新しいプログラム・イメージを実行するように新しい環境を準備しなければならない。そのために、デーモン・アプリケーション１０８はカーネル層１０４にｆｏｒｋ（）システム・コールを出して新しいプロセスを作成する（ステップ１２２）。

【００１５】ｆｏｒｋ（）システム・コールは、カーネル層１０４でカーネルｆｏｒｋルーチン１２４をトリガして制御を獲得させる。ｆｏｒｋルーチン１２４は制御を獲得すると新しいアドレス空間１０６を作成して初期設定し（ステップ１２６）、デーモン層１０２からユーザ・タスク・プロセス層１０６に記憶属性、セキュリティ属性、及びプロセス実行属性をコピーする（ステップ１２８）。

【００１６】正常に完了すると、ｆｏｒｋ（）ルーチン１２４はデーモン・アプリケーション１０８を持つプロセス層１０２が、デーモン・アプリケーション１３０を持つ新たに作成された子プロセス層１０６の親になるようにする。ｆｏｒｋ（）ルーチン１２４から、デーモン・アプリケーション１０８及び１３０にそれらのデーモン・アプリケーションがどのプロセス層で実行されているかを示す標識を返す。親プロセス層１０２内の場合、その層内のデーモン・アプリケーション１０８はループ・バックして、ポート１１０からさらに作業が送られてくるのを待つ（ステップ１２９）。

【００１７】子層１０６内の場合、その層内のデーモン

・アプリケーション１１４は要求で指定されたユーザ名１１４のセキュリティ属性を設定する。

【００１８】いくつかのＰＯＳＩＸ関数が呼び出され、子層１０６の正しい補足グループとグループＩＤとユーザＩＤとが設定される。これらのコールは、セキュリティ・データベース１３４内の要求側ユーザのデータにアクセスする。具体的には、子デーモン・アプリケーション１３０はまずｇｅｔｐｗｎａｍ（）コールを出してセキュリティ・データベース１３４にアクセスし、ユーザ名１１４に対応するユーザＩＤとグループＩＤを判断する（ステップ１３２）。アプリケーション１３０は次にｇｅｔｇｒｏｕｐｓ（）コールを出してセキュリティ・データベース１３４にアクセスし、ユーザ名１１４に対応する補足グループを判断する（ステップ１３８）。

【００１９】この情報を使用して、子デーモン・アプリケーション１３０はｓｅｔｇｒｏｕｐｓ（）コールを出して子層１０６の補足グループをユーザ名１１４に対応する補足グループに設定し（ステップ１４０）、ｓｅｔｇｉｄ（）コールを出して子層１０６のグループＩＤをユーザ名１１４に対応するグループＩＤに設定し（ステップ１４２）、ｓｅｔｕｉｄ（）コールを出して子層１０６のユーザＩＤをユーザ名に対応するユーザＩＤに設定する（ステップ１４４）。

【００２０】上述のようにして新しいユーザのために正しいＰＯＳＩＸセキュリティ環境が設定されると、デーモン・アプリケーション１１４は指定されたユーザ・タスク識別子１１８をパラメータとして使用してｅｘｅｃ（）システム・コールを出す（ステップ１４６）。この関数によってカーネルｅｘｅｃルーチン１４８はアドレス空間１０６を初期設定し直し、記憶域を消去し、正しいプロセス・セキュリティと実行環境を設定し（ステップ１５０）、その後で新しいユーザ・タスク・プログラム１５４に制御を渡す（ステップ１５２）。

【００２１】ＯｐｅｎＥｄｉｔｉｏｎ拡張機能付きのＩＢＭ　ＯＳ／３９０オペレーティング・システムなどの多くのＰＯＳＩＸ準拠システムでは、ユーザ・プロセス内のユーザ・アイデンティティを変更するためにセキュリティ・データベースに対して行われる過度のコールに関する重大なパフォーマンス上の不利がある。また、この従来の方法では、デーモンはＰＯＳＩＸのｓｐａｗｎ（）サービスを利用することができず、したがってデーモン・アプリケーションの一部を子プロセスで実行するようにしなければならない。

【００２２】図２は、本発明を組み込んだコンピュータ・システム２００（物理機械を含む）の例を示す図であり、指定されたユーザＩＤのために要求されたタスクを実行する様々なソフトウェア層間の関係が図示されている。これらの層には、デーモン・プロセス層またはアドレス空間２０２と、オペレーティング・システム（ＯＳ）またはカーネル層またはアドレス空間２０４と、（作成された場合は）ユーザ・タスク・プロセス層またはアドレス空間２０６とが含まれる。

【００２３】デーモン・プロセス層２０２には、遠隔クライアント（図示せず）に結合されたポートまたは通信回線２１０を監視するソフトウェアを含むデーモン・アプリケーション２０８がある。デーモン・アプリケーション２０８は、入力としてユーザＩＤ２１４とパスワード２１６と実行する特定のタスクの識別子２１８とを含むクライアント要求２１２をポート２１０を介して受け入れる（ステップ２２０）。

【００２４】デーモン・アプリケーション２０８は、ユーザＩＤ２１４を検証し、パスワード２１６を認証した後、要求されたタスクを行う新しいプログラム・イメージを実行するように新しい環境を準備する。これを行うために、デーモン・アプリケーション２０８はまず、環境変数ＵＳＥＲＮＡＭＥを処理する要求２１２のユーザ名２１４に設定する（ステップ２２２）。デーモン・アプリケーション２０８は次に、カーネル層２０４に対してｓｐａｗｎ（）システム・コールを出して新しいプロセスを作成する（ステップ２２４）。ｓｐａｗｎ（）システム・コールはパラメータとしてタスク識別子２１８を渡し、環境変数ＵＳＥＲＮＡＭＥとしてユーザ名２１４を渡す。次にデーモン・アプリケーション２０８はルーチンの先頭にループ・バックしてさらに作業を入手する（ステップ２３６）。

【００２５】ｓｐａｗｎ（）システム・コール（ステップ２２４）によって、カーネル層２０４内のカーネルｓｐａｗｎルーチン２２６が制御を獲得する。ｓｐａｗｎルーチン２２６は制御を獲得するとまず新しいユーザ・タスク層またはアドレス空間２０６を作成する（ステップ２２８）。

【００２６】ｓｐａｗｎルーチン２２６は次に、カーネル層２０４内のセキュリティ・データベース２３２に照会して必要なＰＯＳＩＸ許可設定値を入手する（ステップ２３０）。次に、正しいセキュリティ許可を使用して、すなわちデータベース２３２内の（環境変数ＵＳＥＲＮＡＭＥに指定されている）ユーザ名２１４に対応づけられたグループＩＤとユーザＩＤに等しく設定された新しいアドレス空間２０６のユーザＩＤとグループＩＤを使用して、新しいアドレス空間２０６を直接初期設定する。

【００２７】最後に、カーネルｓｐａｗｎルーチン２２６は、要求２１２で指定された（デーモン・アプリケーション２０８によってパラメータとして渡された）ユーザ・タスク２１８に対応するプログラム・イメージ２３８をアドレス空間２０６にロードし、そのプログラム・イメージに制御を渡してユーザ・タスクを実行する（ステップ２３４）。

【００２８】遠隔ユーザによる悪用を避けるために、新しい環境変数ＵＳＥＲＮＡＭＥは許可されたユーザのみ

に制限する必要がある。この機能を使用するのに必要な特権は、ｓｃｔｕｉｄ（）関数と同等でなければならない。（デーモン・アプリケーションは通常そうであるように）デーモン・アプリケーション２０８が無制限のアクセス権を持つスーパーユーザ（ユーザＩＤ＝０）として動作する場合、デーモン・アプリケーション２０８は必要な権限を持つことになる。

【００２９】上述のように、本発明は、デーモン・プロセス環境に変更を加える必要なしに、また、デーモン・アプリケーションのどの部分もユーザ・タスク・プロセス内で実行する必要なしに、ユーザ・タスク・プロセスのためのセキュリティ環境を作成する方法を提供する。また、本発明は、従来のｆｏｒｋ（）関数とｅｘｅｃ（）関数をｓｐａｗｎ（）関数で置き換えることができるようにし、それによってｓｐａｗｎ（）関数が本質的に備える簡略さとパフォーマンス強化を実現する。

【００３０】当業者なら様々な修正がわかるであろう。上記のように本発明についてＵＮＩＸベースのシステム、具体的にはＰＯＳＩＸ準拠システムの文脈で説明したが、本発明はその他のシステムでも使用可能であることは明らかであろう。

【００３１】まとめとして、本発明の構成に関して以下の事項を開示する。

【００３２】（１）要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視するサーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わって前記タスクを実行する方法であって、

（ａ）デーモン・プロセスが、前記ユーザからの前記要求を受け取ると、［１］前記要求で指定された前記アイデンティティに従って環境変数を設定するステップと、［２］前記オペレーティング・システム・カーネルに対してシステム・コールを出して前記指定されたユーザ・タスクを新しいアドレス空間内で実行するステップと、

（ｂ）前記オペレーティング・システム・カーネルが前記デーモン・プロセスから前記システム・コールを受け取ると、［１］前記指定されたユーザ・タスクのために新しいアドレス空間を作成するステップと、［２］前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成するステップと、［３］前記新しいアドレス空間内で前記指定されたユーザ・タスクを開始するステップとを含む方法。

（２）前記オペレーティング・システム・カーネルがＰＯＳＩＸ準拠オペレーティング・システム・カーネルであることを特徴とする、上記（１）に記載の方法。

（３）前記システム・コールがｓｐａｗｎ（）システム・コールであることを特徴とする、上記（１）に記載の方法。

（４）前記ユーザがユーザ名を有し、前記アイデンティティが前記ユーザ名を含むことを特徴とする、上記（１）に記載の方法。

（５）前記環境変数が前記ユーザ名と等しく設定されることを特徴とする、上記（４）に記載の方法。

（６）前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザＩＤを有し、前記セキュリティ環境を作成する前記ステップ（ｂ）［２］が、前記新しいユーザ・アドレス空間のユーザＩＤを、前記環境変数によって指定されたユーザ名のユーザＩＤと等しく設定することを特徴とする、上記（５）に記載の方法。

（７）前記ユーザ名のユーザＩＤを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、上記（６）に記載の方法。

（８）前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループＩＤを有し、前記セキュリティ環境を作成する前記ステップ（ｂ）［２］が、前記新しいアドレス空間のグループＩＤを、前記環境変数によって指定されたユーザ名のグループＩＤと等しく設定することを特徴とする、上記（５）に記載の方法。

（９）要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視する前記サーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わって前記タスクを実行する装置であって、

（ａ）前記デーモン・プロセスに関連づけられ、前記ユーザからの前記要求の受取りに応答して、［１］前記要求に指定された前記アイデンティティに従って環境変数を設定し、［２］前記オペレーティング・システム・カーネルに対してシステム・コールを出して新しいアドレス空間内で前記指定されたユーザ・タスクを実行する手段と、

（ｂ）前記オペレーティング・システム・カーネルに関連づけられ、前記デーモン・プロセスからの前記システム・コールの受取りに応答して、［１］前記指定されたユーザ・タスクのために新しいアドレス空間を作成し、［２］前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成し、［３］前記新しいアドレス空間内で前記指定されたユーザ・タスクを開始する手段とを含む装置。

（１０）前記オペレーティング・システム・カーネルがＰＯＳＩＸ準拠オペレーティング・システム・カーネルであることを特徴とする、上記（９）に記載の装置。

（１１）前記システム・コールがｓｐａｗｎ（）システム・コールであることを特徴とする、上記（９）に記載の装置。

（１２）前記ユーザがユーザ名を有し、前記アイデンティティが前記ユーザ名を含むことを特徴とする、上記（９）に記載の装置。

（１３）前記環境変数が前記ユーザ名と等しく設定されることを特徴とする、上記（１２）に記載の装置。

（１４）前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザＩＤを有し、前記セキュリティ環境を作成する前記手段（ｂ）［２］が、前記新しいアドレス空間のユーザＩＤを、前記環境変数によって指定されたユーザ名のユーザＩＤに等しく設定する手段を含むことを特徴とする、上記（１３）に記載の装置。

（１５）前記ユーザ名のユーザＩＤを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、上記（１４）に記載の装置。

（１６）前記アドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループＩＤを有し、前記セキュリティ環境を作成する前記ステップ（ｂ）［２］が、前記新しいアドレス空間のグループＩＤを、前記環境変数によって指定されたユーザ名のグループＩＤに等しく設定することを特徴とする、上記（１３）に記載の装置。

（１７）要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・システム・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視する前記サーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わってタスクを実行する方法ステップを行う機械によって実行可能なプログラムを有形に実施する機械可読プログラム記憶装置であって、前記方法ステップが、

（ａ）前記デーモン・プロセスが、前記ユーザからの前記要求を受け取ると、［１］前記要求に指定された前記アイデンティティに従って環境変数を設定するステップと、［２］前記オペレーティング・システムに対してシステム・コールを発行して前記指定されたユーザ・タスクを新しいアドレス空間内で実行するステップと、

（ｂ）前記オペレーティング・システム・カーネルが、前記デーモン・プロセスから前記システム・コールを受け取ると、［１］前記指定されたユーザ・タスクのために新しいアドレス空間を作成するステップと、［２］前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成するステップと、［３］前記指定されたユーザ・タスクを前記新しいアドレス空間内で開始するステップとを含む、プログラム記憶装置。

（１８）前記オペレーティング・システム・カーネルがＰＯＳＩＸ準拠オペレーティング・システム・カーネルであることを特徴とする、上記（１７）に記載のプログラム記憶装置。

（１９）前記システム・コールがｓｐａｗｎ（）システム・コールであることを特徴とする、上記（１７）に記載のプログラム記憶装置。

（２０）前記ユーザがユーザＩＤを有し、前記アイデンティティが前記ユーザＩＤを含むことを特徴とする、上記（１７）に記載のプログラム記憶装置。

（２１）前記環境変数が前記ユーザＩＤと等しく設定されることを特徴とする、上記（２０）に記載のプログラム記憶装置。

（２２）前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザＩＤを有し、前記セキュリティ環境を作成する前記ステップ（ｂ）［２］が、前記新しいアドレス空間のユーザＩＤを、前記環境変数によって指定されたユーザ名のユーザＩＤと等しく設定するステップを含むことを特徴とする、上記（２１）に記載のプログラム記憶装置。

（２３）前記ユーザ名の前記ユーザＩＤを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、上記（２２）に記載のプログラム記憶装置。

（２４）前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループＩＤを有し、前記セキュリティ環境を作成する前記ステップ（ｂ）［２］が、前記新しいアドレス空間のグループＩＤを、前記環境変数によって指定されたユーザ名のグループＩＤと等しく設定するステップを含むことを特徴とする、上記（２１）に記載のプログラム記憶装置。

【図面の簡単な説明】

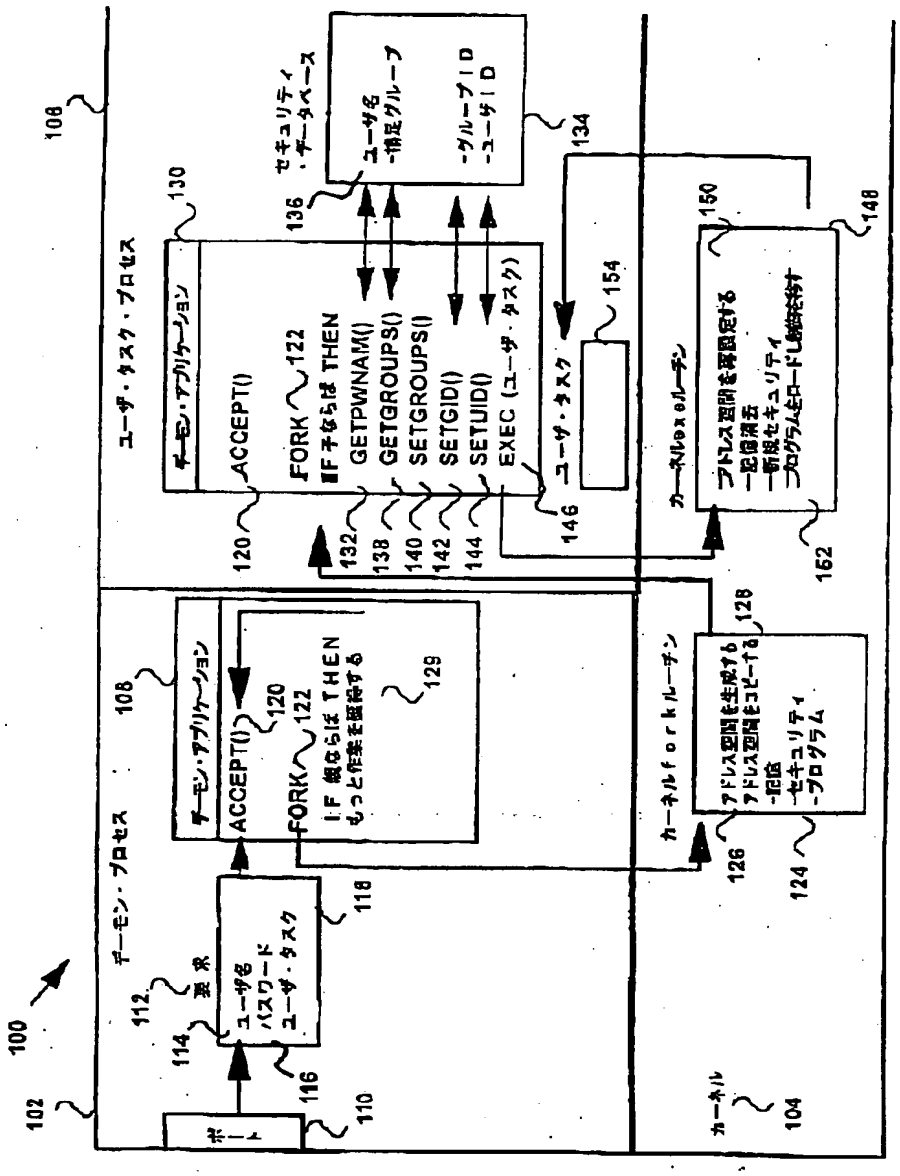
【図１】ユーザ要求タスクのためのセキュリティ環境を作成する従来の実施態様を組み込んだコンピュータ・システムを示す図である。

【図２】ユーザ要求タスクのためのセキュリティ環境を作成する本発明の組込みを使用したコンピュータ・システムを示す図である。

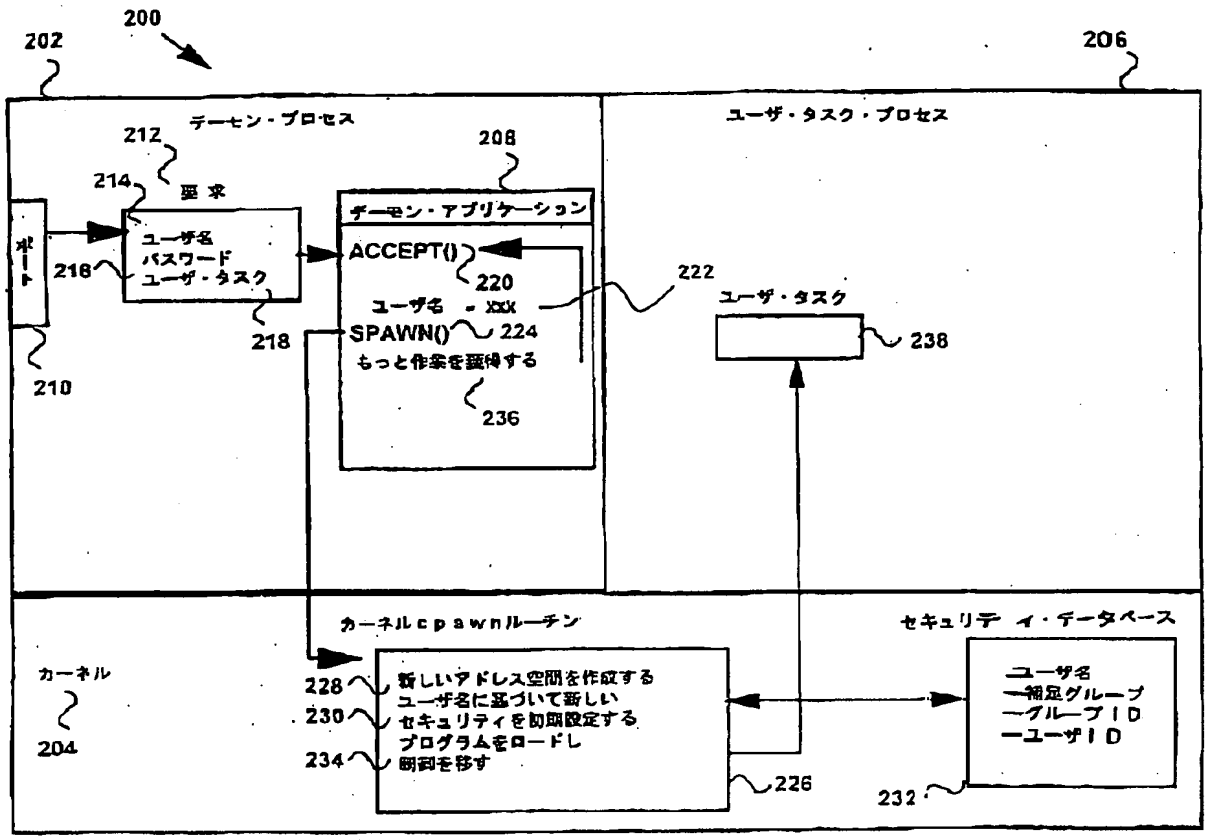
【符号の説明】

２００　コンピュータ・システム
２０２　デーモン・プロセス層
２０４　カーネル層
２０６　ユーザ・タスク・プロセス層
２０８　デーモン・アプリケーション
２１０　ポート
２１４　ユーザ名
２２６　カーネルｓｐａｗｎルーチン
２３２　セキュリティ・データベース
２３８　ユーザ・タスク

【図１】


100
102
104
106
108
110
112
114
116
118
120
122
124
126
128
129
130
132
134
136
138
140
142
144
146
148
150
152
154
デーモン・プロセス
ユーザ・タスク・プロセス
カーネル
ブート
要求
ユーザ名
パスワード
ユーザ・タスク
デーモン・アプリケーション
ACCEPT()
FORK
IF 偽ならば THEN
もっと作業を継続する
カーネルforkルーチン
アドレス空間を生成する
アドレス空間をコピーする
-属性
-セキュリティ
-プログラム
セキュリティ・データベース
ユーザ名
-補足グループ
-グループID
-ユーザID
GETPWNAM()
GETGROUPS()
SETGROUPS()
SETGID()
SETUID()
EXEC (ユーザ・タスク)
ユーザ・タスク
カーネルexecルーチン
アドレス空間を再設定する
-記憶消去
-新規セキュリティ
プログラムとヒープ領域を設定

【図2】



フロントページの続き

(72)発明者　アーネスト・スコット・ベンダー
アメリカ合衆国12477　ニューヨーク州ソージャーティーズ　パイン・グローブ・スクール・ロード　27

(72)発明者　マイケル・ゲイリー・スピーゲル
アメリカ合衆国10950　ニューヨーク州モンローサンセット・ハイツ　10